UCRL-MA-118785

The JCM Terrain Builder Guide

Version 2.2

Jeff Pimper
Mike Uzelac

September 1994

Lawrence
Livermore
National
Laboratory

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

The JCM Terrain Builder Guide

Version 2.2

September 1994

Contents

Figures

Introduction

The JCM Terrain Builder (TERBUILD) program allows the user to create JCM 2.2 format terrain files from raw, digitized elevation data files in several different formats. These include the Defense Mapping Agency's 100 meter data (UTM100) and Digitized Terrain Elevation Data (DTED) formats, Atmospheric Release Advisory Committee's (ARAC) formatted satellite data, four different Janus formats, JCM 1 format and JCM 2.2 files themselves. It is also possible for users to extend the program to include their own special formats. Detailed descriptions of each of the terrain formats which are supported by the Terrain Builder are included in the Appendix to this document.

JCM terrain is stored as an array of terrain cells. Information about the area represented by the cell, such as tree or building density or the presence or absence of roads, is contained in the cell. The cell also contains the elevation at the location represented by the lower left corner of the cell. The Terrain Builder program fills in the elevation field of each cell. TEREDIT, the JCM Terrain Editor, is used to fill in the cities and trees, rivers and roads.

System Logical Names

The Terrain Builder program uses some logical names as pointers to files, directories, and devices. These names are:

CONVFORM - Points to the directory which contains the TERBUILD FMS forms library.

TAPEFILE - Points to the system nine-track tape drive, this logical name is only needed
if tapes are to be processed.

TERRAIN - Points to the directory which contains the standard JCM terrain files (files 1
- 49).

These names must be ASSIGNed or DEFINEd before the Terrain Builder may be executed.

Forms Management System

The Terrain Builder makes heavy use of the DEC Forms Management System (FMS). FMS provides two important features; is allows all user input to be checked for validity, and it includes Help features. If at any time the user enters invalid data, then an error message is displayed at the bottom of the screen. If at any time the user is unsure of what to do next, he can get Help by pressing the {**PF2**} key on the keypad. This will usually cause a one line help message to appear at the bottom of the screen. Additional Help can often be displayed by pressing {**PF2**} a second time.

As TERBUILD progresses, different FMS forms are displayed on the terminal. These forms contain explanatory text, user prompts, and input fields. TERBUILD asks for user input by displaying a form and then moving the screen cursor to the desired input field, which is then under-lined and placed in bold type to indicate that input from the user is desired. If there are several fields on the form, the user can move the cursor from one field to another by using the **TAB** and **BS** {**F12**} (backspace) keys. Data is entered in fields in one of three ways:

1. If the cursor is sitting at the leftmost position in the field, then TERBUILD is expecting textual input, such as a file name. Type in the requested information and press **TAB**, **BS** {**F12**}, or **RETURN**. If you make a mistake, use the **DELETE** key to back over and delete the mistake and then retype the correct text. This is called a left-justified field.

2. If the cursor is sitting past the rightmost position of the field, then as data is entered it appears to the left of the cursor. Every time a new character is entered, the contents of the field are moved one space to the left and the new character appears at the rightmost position in the field. Pressing the **DELETE** key causes the entire field to be moved one position to the right, deleting the rightmost character. This is called a right-justified field and is used mainly to enter numbers and commands.

3. The third type of field is a combination of the two previous types and is used to enter floating point numbers, i.e. numbers with a decimal point. When this type of field is displayed, the cursor appears on top of the decimal point. Digits to the left of the decimal point are entered as if they were part of a right-justified field. To enter digits to the right of the decimal point, first press the decimal point on the keyboard and then enter the digits as if they were in a left-justified field.

Most functions may be aborted by pressing the {**PF4**} key. There is usually a prompt on the screen to indicate whether or not the {**PF4**} key may be used.

Running the Terrain Builder

Use the standard DCL Run command to invoke the JCM Terrain Builder or initiate it from the Top Level JCM menu (Fig. 1) by selecting option **A6**. The program begins by displaying its copyright notice. Press either the **RETURN** or the **ENTER** keys to proceed to the Terrain Builder Main menu (Fig. 2).

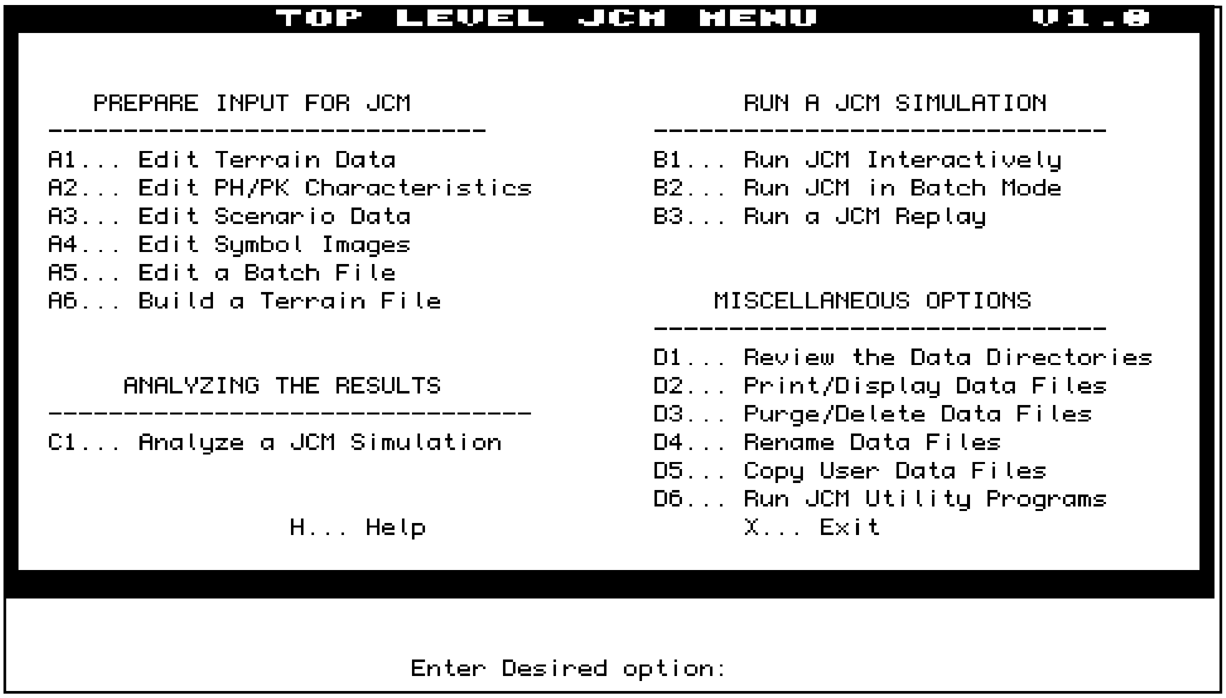

Figure 1 Top Level JCM menu

Main Menu

Terrain Builder Commands

T. **Transfer Files from Tape to Disk**

I. **Initialize a new Terrain File**

R. **Read a Terrain File from Disk**

W. **Write Current Terrain File from Disk**

C. **Convert Raw Data to JCM format**

D. **Direct File Conversion**

L. **Convert Lat/Long to UTM**

X. **eXit TERBUILD**

Enter Choice:

Figure 2 Terrain Builder Main menu

Note: The **W** option, Write Current Terrain File from Disk, only appears if a new terrain file has been initialized or if a previously created file has been read in.

The **C** option only appears if the current terrain file in memory still has areas which have not been filled in with elevation data.

To perform any of the menu functions, the user enters one of the letters in the left hand column.

Transfer Files from Tape to Disk

If several terrain files are to be created from a tape file, it is often faster and more convenient to first transfer the data from the tape to a disk file and then use the disk file for the creating process. This function allows the user to transfer DTED files from tape to disk. It also has a "hook" which allows the user to include his own tape to disk routine.

The transfer function is invoked from the main menu by entering **T** or the function choice and pressing the **RETURN** key. The Transfer menu will then be displayed:

Transfer Tape to Disk

Select File Type

A. DTED format

U. User defined format

R. Return to Main Menu

Format:

Figure 3 Transfer menu

First mount the tape on a tape drive. The logical name TAPEFILE should be assigned to the tape drive. This is normally done by the JCM shell, however if TERBUILD is run in a standalone mode then this logical name must be defined before TERBUILD is started.

Secondly, select the menu code for the appropriate tape format; **A** for DTED format, or **U** for a user defined format. The tape will be mounted and the tape and file header information will be displayed on the terminal, along with the question:

Do you wish to transfer this tape file?

If the answer is Yes, then enter **Y**. TERBUILD will then ask for the name of the file to be created:

ENTER Disk File Name:

This function may be aborted at this time by pressing the {**PF4**} key. If a file name is entered, then the file will be read from the tape and written to the indicated disk file. If the entered file name does not have an extension then the extension .DAT is automatically appended to the name. As the transfer process proceeds, a record count is displayed at the bottom of the screen.

When the transfer is complete, or if the answer to the first question was **N** for No, a new question is displayed:

Do you wish to continue with the next file on this tape?

If the answer is **N** for no, then the tape is dismounted and the Transfer menu is redisplayed. If the answer is **Y** for yes, then the tape is advanced to the next file. If there is no next file, then the tape is dismounted and the transfer menu is redisplayed. If there is another file, then its file header is displayed and the user is asked if he wants to transfer it.

This process may be repeated in order to read several files from a tape. The current tape may be dismounted and a new tape mounted any time the Transfer menu is being displayed on the terminal.

ENTER "R" to RETURN to the main menu.

Initialize a New Terrain File

The **I** option from the Main menu is used to create and initialize a new JCM 2.2 terrain file. In order to create a new file, the user must indicate the location of the lower left corner of the file, the width and height of the file (in kilometers), and the desired terrain grid resolution (Fig. 4).

Initialize UTM File

Initialization Commands:

S. Standard UTM file

P. Pseudo UTM File

X. eXit and do nothing

Enter Choice:

Figure 4 Initialization menu

When the desired values have been entered, press **RETURN**. If there already is a terrain file in memory, then the user is asked:

Do you wish to save the terrain file currently in memory?

If the answer is yes then he is asked:

Output Terrain File Number?

The user may specify a file number in the range 50 - 99 or press the {**PF4**} key to abort the save. If a number was entered, the file is saved.

Universal Transverse Mercator (UTM) zones are six degrees of Longitude wide, starting at zero degrees East or West. If the desired terrain area does not cross a UTM zone boundary (6, 12, 18...84 etc.) then the terrain file can be initialized as a "Standard UTM file" option **S**. If the terrain area does cross a UTM boundary then the file **MUST** be initialized as a "Pseudo UTM file" option **P**. Because of the earth's curvature, North/South UTM Grid lines are not parallel at the six degree zone boundaries. Because JCM only displays parallel grid lines, if a JCM terrain file crosses a zone boundary a new (pseudo) set of North/South grid lines is created in which the lines are parallel. The North/South grid lines on a pseudo JCM terrain file will not match the grid lines shown on a map. The East/West lines and Lat/Long grid lines will be correct. The "LOCATE" functions available on the JCM menu will display the pseudo UTM, real UTM and Lat/Long coordinates of a selected point on a pseudo terrain. After selecting either option the following menu will appear:

Initialize a New File
with standard UTM coordinates

	Easting	Northing	Grid Zone
Lower Left Corner (UTM Kilometers):	**0.000 E**	**0.000 N**	**0**
File Width and Height (Kilometers):	**50.000 x 50.000**		
Terrain Grid Resolution:	**500 x 500 cells**		

Description of terrain file area:

Hit RETURN when finished
PF4 aborts this function

PF1-PF4 to switch to Latitude/Longitude format

Figure 5 Coordinates menu

To abort this function without initializing a new file, press the {**PF4**} key and the Main menu (Fig. 2) will be redisplayed.

The terrain cells in a newly initialized file contain a special value which is interpreted by the Terrain Builder to mean null or no value. If a terrain file containing null values is read by TEREDIT, the JCM Terrain Editor, the cells containing null values are displayed as density seven cities occupying a terrain cell with an elevation at its lower left corner of 65535 meters. It is usually very obvious to the eye when such a terrain file is displayed by TEREDIT.

Lower Left Corner

The lower left corner of the file may be entered in either of two ways: UTM coordinates or Latitude and Longitude. The default coordinate system is the UTM grid system.

UTM coordinates are expressed as the distances, in kilometers, from two meridian lines. The base meridian for North/South coordinates is the Equator. The base meridian for East/West coordinates varies based on which UTM zone is being used. The typical map, as generated by the Defense Mapping Agency, shows the coordinate system being used along the edges of the map. The coordinates are shown in an unusual format; the first few digits are printed in a very small type font, then there are two digits in a large type font, then three more digits in the small type font again. Often the last three digits are skipped, but the leftmost digits and the large digits are always there. To convert these coordinates to kilometers, place a decimal point to the right of the large digits and read the number.

The military uses a shorthand version of the system which eliminates some of the digits from the coordinate numbers. First they throw away the leftmost digits, the ones in the small type font, so their coordinates always start with the two large type font digits. The rest of the coordinate depends on how accurate the coordinate needs to be. Typical coordinates are expressed as six, eight, or ten digits. The East/West part of the coordinate is the left half of the number, the North/South part is the right half. For example, the eight digit coordinate 12345678 is interpreted as 12.34 East/West, 56.78 North/South and is accurate to the nearest 10 meters. A six digit coordinate, 123456, is interpreted as (12.3, 45.6) and is accurate to the nearest 100 meters. A ten digit coordinate, 1234567890, is interpreted as (12.345, 67.890) and is accurate to the nearest meter.

When TERBUILD asks for UTM coordinates, the user **MUST** enter the full coordinate value, including all the leftmost digits and with a decimal point after the two large type font digits. Coordinates may be entered down to one meter accuracy if desired.

The lower/left corner may also be entered in Latitude and Longitude. To switch back and forth from UTM to Lat/Long, first press the {**PF1**} key and then press the {**PF4**} key. Lat/Long coordinates are entered in the usual format, i.e. degrees : minutes : seconds followed by one of the letters "N" or "S" for North or South Latitude or "E" or "W" for East or West Longitude.

The Lat/Long entered is translated into UTM coordinates for JCM.

UTM grid zones are six degrees wide, starting at zero degrees East or West. The Grid Zone can be calculated from the Longitude of the lower left corner of the terrain area.

1. Divide the Longitude by six.
2. Throw away the part to the right of the decimal point.
3. Multiply the result by six.
4. Add three

For example if the Longitude is 83 degrees West then:

1. Divide by six (83 / 6 => 13.8333)
2. Throw the fractional part away (13.8333 = 13)
3. Multiply by 6 (13 * 6 = 78)
4. Add three (78 + 3 = 81)

Terrain Width and Height

Once the lower left corner is specified the user specifies the "width" and "height" of the file. "Width" refers to the distance from the West edge to the East edge, "height" refers to the distance from the South edge to the North edge. The distances are expressed in kilometers. It is not required that the width and height be the same. However if they are not then the graphic terrain display created by TEREDIT will be warped because it always displays the entire file in a square window.

Terrain Grid Resolution

This part of the initialization allows the user to specify the resolution of the terrain file to be created. Digitized terrain files represent real terrain by creating a mesh or gridwork of equally spaced data points. The real world is sampled at each data point and the elevation at that point is stored in the terrain file. Whenever an elevation is required by JCM 2.2 it is calculated by interpolating from among the four grid points which surround the point in question. The closer the grid points are, the more accurate the interpolated value.

The JCM 2.2 terrain grid currently allows for a maximum of 1000 by 1000 cells. The horizontal (East/West) resolution of a file is calculated by dividing the width of the file in meters by the width of the file in grids. The vertical (North/South) resolution of the file is calculated by dividing the height of the file in meters by the height of the file in grids. For example, if the file is 100 by 100 kilometers and 1000 by 1000 grids, then the resolution is 100,000m/1000 or 100 meter resolution.

It is not required that the terrain have the same number of grids in both the vertical and horizontal direction. The JCM terrain handler automatically compensates for terrain files which are not square or which have differing horizontal and vertical resolutions.

Description of Terrain File Area

The TERRAIN BUILDER enables the user to add an eighty character descriptor to each terrain file. This descriptor can be used to identify the file with information such as where is the area, where the data came from and who made the file. This data appears when the terrain file is used by the *Terrain Editor*. After the descriptor is entered the intialization is complete and the user returns to the main menu by pressing **RETURN.**

Read a Terrain File from Disk

This option is used to read files back into memory so that more processing may be performed on them. This function is invoked by entering the **R** command on the Main menu (Fig. 2) and pressing **RETURN**.

If there already is a terrain file in memory, then the user is asked:

Do you wish to save the terrain file currently in memory?

If the answer is yes then the user is asked:

Output Terrain File Number?

The user may specify a file number in the range 50 - 99 or press the {**PF4**} key to abort the save. If a number is entered, the file is saved.

The user is then asked:

Input Terrain File Number?

The user may specify a file number in the range 50 - 99 or press the {**PF4**} key to abort the read. If a number is entered, the requested file is read, replacing any file currently in memory. After the file has been read in, it is checked to see if there are any null-values in it. If there are no null valued terrain cells the user is informed that the terrain file has been completed and he is asked to enter another terrain file number.

If the terrain file has null-valued terrain cells then some internal pointers and parameters are set and the Main menu is redisplayed to continue processing.

Write Current Terrain File to Disk

The Terrain Builder allows the user to save the current terrain file at any time, whether or not all the null cells have been filled in. This option is invoked from the Main menu (Fig. 2) by entering the command **W** and pressing **RETURN**.

The user is then asked:

Output Terrain File Number?

The user may specify a file number in the range 50 - 99 or press the {**PF4**} key to abort the save. If a number was entered, then a copy of the current terrain file is saved to disk. When the save is finished, the Main menu is redisplayed. The current file is still in memory and further processing may be performed on it.

The Write option may only be invoked when there is a terrain file in memory.

Convert Raw Data to JCM Format

This function takes raw elevation data, converts it, and fills in the null valued terrain cells in the terrain file being created. It is invoked by entering the **C** command from the Main menu (Fig. 2) and pressing **RETURN**. The Convert menu will be displayed:

Convert Raw Data to JCM Format

Select File Type

A. **ARAC format**
B. **DTED Tape format**
C. **DTED CD format**
D. **Janus 4.X format**
E. **Janus 5.1 format**
F. **Janus (T) format**
G. **Janus (A) format**

```
H.    JCM 1.X format
I.    JCM 2.X format

U.    User defined format



R.    Return to Main Menu

      Format:
```

Figure 6 Convert menu

Options **A** through **I** are used to process raw data files in one of the standard formats. Option **U** is used to process user defined files. However, in order to use option **U**, the user must also create the necessary subroutines to do the processing, compile them, add them to the TERBUILD library, and then relink the TERBUILD program.

ARAC Format

The data format for ARAC files is described in Appendix A.

This option is invoked from the Convert menu by entering the **A** command and pressing **RETURN**. The descriptors for the file being created are displayed at the top of the screen. The number of terrain cells still containing null values is displayed at the bottom of the screen (Fig. 7).

```
    Description of JCM 2.x format terrain file being built:
The eight character descripter appears on this line.

  UTM UR Corner:     0.000 0.000    LL Latitude:    0 00” 0.00 N
  UTM LL Corner:     0.000 0.000    LL Longitude    0 00” 0.00 E

  Terrain Grid Resolution:    1000 x 1000 cells



                   Enter Disk File Name:
```

There are 100000 undefined terrain cells.

Figure 7 Conversion menu

Enter the name of the ARAC disk file to be processed. The default extension of .DAT is applied to the file name if no extension was entered. To abort this function, press the {**PF4**} key.

The ARAC file is read and its descriptors are displayed on the screen. If the ARAC file does not overlap the terrain file being created a message is displayed on the screen. Press **RETURN** and TERBUILD will ask for another ARAC file.

If the ARAC file overlaps the terrain file being created, then data is transferred from the ARAC file to the terrain file. As the ARAC file is being processed, a scanline counter and the number of terrain cells left to fill is displayed at the bottom of the screen.

If, once the ARAC file has been completely processed, the terrain file is not finished, TERBUILD will ask:

Is there another ARAC file?

If the answer is **Y**, then you will be given an opportunity to save the current status of the new file:

Output Terrain File Number?

Enter a number in the range 50-99 to save the file or press {**PF4**} to skip the save. You will then be asked to enter the file name of the next ARAC file to be processed and the conversion will be repeated.

If there is not another ARAC file, then you will be given the opportunity to manually finish off the terrain file. If the file is still unfinished then the Conversion menu will be redisplayed.

If the terrain file is complete then TERBUILD will display:

Finishing off the terrain file

When complete you will be asked if you want to save the terrain file:

Output Terrain File Number?

Again, enter a number in the range 50-99 to save the file or press {**PF4**} to skip the save. When finished the Main menu will be redisplayed.

DTED Tape Format

The data format for DTED files is described in Appendix A.

This option is invoked from the Convert menu by entering the **B** command and pressing **RETURN**. The descriptors for the file being created are displayed at the top of the screen. The number of terrain cells still containing null values is displayed at the bottom of the screen.

TERBUILD asks:

Is the Source Data on Tape or Disk?

Enter **T** for tape, **D** for disk.

If the source data is on disk then you will be asked to enter the name of the disk file:

ENTER Disk File Name:

Once you enter a name, file processing continues in a manner similiar to that for ARAC files. The only difference being that you may be asked to enter calibration points for the conversion from Latitude and Longitude to the UTM coordinate system.

If the source data is on tape then you will be asked:

Has the tape been loaded on the tape drive?

Enter **Y** and press the **RETURN** key when the tape drive is ready.
Enter **N** and press the **RETURN** key to abort this function.

TERBUILD reads the tape header label file and displays the tape labels, including the file descriptors from the user label. You may be asked to enter Latitude/Longitude to UTM calibration points. Once the file is calibrated, TERBUILD will check to see if the file on tape contains the information needed to created the new terrain file. If not, then you will be notified and asked if you want to continue with the next file on the tape (if any):

Do you wish to continue with the next DTED file on this tape?

If the answer is **N** for no, then the tape will be dismounted and TERBUILD will give you an opportunity to finish off the file manually and to save the file onto disk. When done the Convert menu will be redisplayed.

If the answer is **Y** for yes, then the tape will be forwarded to the next tape file and its label information displayed.

If the tape file does contain data which may be used to create the new terrain file then TERBUILD will ask:

Do you wish to process this DTED file?

If you enter **N** and press the **RETURN** key then the file will be skipped and you will be asked if you want to continue with the next file (see above). If you enter **Y** and press the **RETURN** key then the tape file will be read and the data transferred to the terrain file in a manner similiar to data being read from disk files (see above).

In either case (disk or tape) the file will be read until either the new terrain file is finished or until the input file has been exhausted. If the terrain file is not complete then you will be asked for more input.

If the terrain file is complete then TERBUILD will display:

Finishing off the terrain file

When complete you will be asked if you want to save the terrain file:

Output Terrain File Number?

Again, enter a number in the range 50-99 to save the file or press {**PF4**} to skip the save. When finished the Main menu will be redisplayed.

DTED CD Format

The data format for DTED CD files is described in Appendix A.

This option is invoked from the Convert menu by entering the **C** command and pressing **RETURN**. This option allows CD files to be read from disk only, not directly from a CD. This requires that the CD files must be copied from the CDs and placed in the user's [JCMuu.DATAvv.TERRAIN] directory before running the Terrain Builder. When copying the files it is possible that their file characteristics may be changed making them unreadable by the Terrain Builder. When the files are examnied with the DCL command DIR/FULL "filename.dted", they should have listed "Record Format: Fixed Length 512 byte records." If they have a different Record Format the files must be converted before they can be read by the Terrain Builder. A utility program

FILE.EXE has been created that can be used to convert the files. Users having problems with CD file formats should contact LLNL for more information about using this utility.

The descriptors for the file being created are displayed at the top of the screen. The number of terrain cells still containing null values is displayed at the bottom of the screen.

TERBUILD asks:

Enter Disk File Name:

Janus 4.x Format

The data format for Janus 4.X files is described in Appendix A. This option allows you to create new JCM terrain files as subsets or supersets of already existing Janus 4.X files.

This option is invoked from the Convert menu by entering the **D** command and pressing **RETURN**. The descriptors for the file being created are displayed at the top of the screen. The number of terrain cells still containing null values is displayed at the bottom of the screen.

TERBUILD asks:

Read Janus 4.x x Format

Terrain File number ?

Enter a number in the range 1-49 to use a Janus system terrain file or a number in the range 50-99 to use a user terrain file. Pressing {**PF4**} aborts the function.

If you entered a file number, the file is read in and checked to see if it overlaps the area of the file being created. If not, a message is displayed and you are asked to enter a new file number. If the Janus file does overlap the new file, then data is transferred from the Janus file to the new file.

If, when the transfer has finished, the new terrain file is complete then TERBUILD will display:

Finishing off the terrain file

When complete you will be asked if you want to save the terrain file:

Output Terrain File Number?

Again, enter a number in the range 50-99 to save the file or press {**PF4**} to skip the save. When finished the Main menu will be redisplayed.

If the terrain is not complete then you are asked:

Is there another Janus file?

If there is another file to be used, enter **Y** and press **RETURN**. You will be given an opportunity to save the file in progress and then asked for the next Janus file to use. If there is not another file, enter **N** and press **RETURN**. You will be given the oppportunity to manually finish off the file and then to save the file onto disk. After the save request, the Convert menu is redisplayed.

Janus 5.1x Format

The data format for Janus 5.1X files is described in Appendix A. This option allows you to create new JCM terrain files as subsets or supersets of already existing Janus 5.1X files.

This option is invoked from the Convert menu by entering the **E** command and pressing **RETURN**. The descriptors for the file being created are displayed at the top of the screen. The number of terrain cells still containing null values is displayed at the bottom of the screen.

TERBUILD asks:

Read Janus 5.1x / JCM 1.x Format ?

Enter a number in the range 1-49 to use a Janus system terrain file or a number in the range 50-99 to use a user terrain file. Pressing {**PF4**} aborts the function. From this point on TERBUILD behaves exactly the same as if you had read in a Janus 4.x file .

Janus(T) and Janus(A) Format

These options allow you to create new JCM 2.2 terrain files from TRAC Janus(T) or Janus(A) terrain files. These files come in several different formats depending on the version number and the Terrrain Builder may or may not be able to read the terrain files.

The original format is a 200 by 350 cell terrain file. A more recent version uses 350 by 350 cells. There is also a 698 by 698 format which is used by TRAC as a master file from which smaller files are extracted. Other Janus(T) file formats may also be available that can't be read by the Terrain Builder. Appendix A has a more detailed description of each of these formats and how to tell which kinds you may have.

This option is invoked from the Convert menu by entering the **F** or **G** command and pressing **RETURN**. The descriptors for the file being created are displayed at the top of the screen. The number of terrain cells still containing null values is displayed at the bottom of the screen.

TERBUILD asks:

Read Janus(T) Data

Janus(T) Terrain File Number?

ENTER a number in the range 1-999. Pressing {**PF4**} aborts the function. After the file number you need to enter the file type:

JCM(T) Terrain File Type?

Enter **1** for a 200 by 350 cell file, **2** for a 350 by 350 cell file, or **3** for a 698 by 698 cell file. Again, the {**PF4**} key may be used to abort this function.

From this point on TERBUILD behaves exactly the same as if you had read in a JCM 2.x file .

JCM 1.X Format

The data format for JCM 1.X files is described in Appendix A. This option allows you to create new JCM terrain files as subsets or supersets of already existing JCM 1.x files.

This option is invoked from the Convert menu by entering the **H** command and pressing **RETURN**. The descriptors for the file being created are displayed at the top of the screen. The number of terrain cells still containing null values is displayed at the bottom of the screen.

TERBUILD asks:

Read JCM 1.x Format

Terrain File number ?

Enter a number in the range 1-49 to use a JCM system terrain file or a number in the range 50-99 to use a user terrain file. Pressing {**PF4**} aborts the function.

If you entered a file number, the file is read in and checked to see if it overlaps the area of the file being created. If not, a message is displayed and you are asked to enter a new file number. If the JCM file does overlap the new file, then data is transferred from the JCM file to the new file.

If, when the transfer has finished, the new terrain file is complete then TERBUILD will display:

Finishing off the terrain file

When complete you will be asked if you want to save the terrain file:

Output Terrain File Number?

Again, enter a number in the range 50-99 to save the file or press {**PF4**} to skip the save. When finished the Main menu will be redisplayed.

If the terrain is not complete then you are asked:

Is there another JCM file?

If there is another file to be used, enter **Y** and press **RETURN**. You will be given an opportunity to save the file in progress and then asked for the next JCM file to use. If there is not another file, enter **N** and press **RETURN**. You will be given the oppportunity to manually finish off the file and then to save the file onto disk. After the save request, the Convert menu is redisplayed.

JCM 2.X Format

The data format for JCM 2.2 files is described in Appendix A. This option allows you to create new JCM terrain files as subsets or supersets of already existing JCM 2.X files.

This option is invoked from the Convert menu by entering the **I** command and pressing **RETURN**. The descriptors for the file being created are displayed at the top of the screen. The number of terrain cells still containing null values is displayed at the bottom of the screen.

TERBUILD asks:

Read JCM 2.x Format

Terrain File number ?

Enter a number in the range 1-49 to use a JCM system terrain file or a number in the range 50-99 to use a user terrain file. Pressing {**PF4**} aborts the function.

From this point on TERBUILD behaves exactly the same as if you had read in a JCM 1.x file.

User Format Files

This is the "hook" which allows the user to process files of any format not already provided by TERBUILD. The user must provide a routine called CONVERT_USER_FILE and relink the Terrain Builder program.

Finishing a File

TERBUILD copies data from source data files into the new terrain file. If all of the null valued cells in the terrain file have been filled in, then TERBUILD automatically terminates any transfer functions and then performs some additional calculations which complete the terrain file. At this stage the terrain file is ready to be processed by the Terrain Editor, TEREDIT. TEREDIT creates the graphic display used by JCM and also allows the user to add cities, trees, rivers, and roads if desired.

Occassionally the source data files will have holes in them where data is missing or not available. TERBUILD allows the user to repair these holes by filling in the missing data in one of three ways. Whenever this function is invoked a submenu is displayed:

```
There are  nnnnnnn  undefined  cells.

A.     Fill in by interpolation
B.     Fill in manually
C.     Fill in with zeroes


X.     Terminate Fill-In

  Command?
```

Figure 8 Finishing a File menu

Option **A** will cause TERBUILD to examine each undefined cell and try to interpolate a value for it from the cells surrounding it. If its neighboring cells are also undefined, then no interpolation can be done and the cell is left undefined.

Option **B** will cause TERBUILD to locate each undefined cell and query the user what elevation to use for its lower left corner. It does this by displaying the UTM coordinates of the lower left corner and asking for the elevation at that point:

Elevation at nn.nnn, nn.nnn: xxxx meters

(PF4 to skip point, PF1-PF4 to abort manual fill)

The UTM coordinate is (nn.nnn,nn.nnn). Enter the value in the xxxx field. Press the {**PF4**} key to skip the current point and proceed to the next one. Press the {**PF1**} key followed by the {**PF4**} key to abort this manual function.

Option **C** will cause all undefined cells to be filled in with an elevation of zero meters.

The **X** command terminates the fill-in function.

Direct File Conversion

This option is used to do a direct file conversion from one of the specified formats to JCM 2.2 format. The JCM 2.2 file will cover exactly the same area as the source file. If the resolution of the source file is less than 1000 by 1000 cells then the JCM 2.2 file will be created by a one to one copying of the source file cells to the JCM 2.2 file, hence the JCM 2.2 file will have exactly the same resolution as the source file.

If the source file is bigger than 1000 by 1000 cells, then the JCM file is created by taking only selected points. For example, if the source file is bigger than 1000 by 1000 but less than 2000 by 2000, then every other point is taken; if the source file is bigger than 2000 by 2000 but less than 3000 by 3000 then every third point will be chosen; and so on.

This method is used to minimize the errors inherent in interpolation. JCM already interpolates within terrain cells. This avoids interpolating from among values which have themselves been created by interpolation.

The direct file conversion menu looks like this:

Convert Raw Data to JCM Format

Select File Type

A. ARAC format
B. Janus(T) format
C. Janus(A) format

U. User defined format

R. Return to Main Menu

Format:

Figure 9 Direct File Conversion menu

ARAC Format

This option converts an ARAC file into a JCM file. ARAC files are 399 by 399 cells, so the JCM file will be an exact copy of the ARAC file.

The data format for ARAC files is described in Appendix A.

This option is invoked from the Direct File Conversion menu by entering the **A** command and pressing **RETURN**. TERBUILD asks:

ENTER Disk File Name:

Enter the name of the ARAC disk file to be processed. The default extension of .DAT is applied to the file name if no extension was entered. To abort this function, press the {**PF4**} key.

The ARAC file is read and its descriptors are displayed on the screen. As the ARAC file is being processed, a scanline counter and the number of terrain cells left to fill is displayed at the bottom of the screen.

Once the ARAC file has been completely processed TERBUILD will display:

Finishing off the terrain file

When complete you will be asked if you want to save the terrain file:

Output Terrain File Number?

Again, enter a number in the range 50-99 to save the file or press {**PF4**} to skip the save. When finished the Main menu will be redisplayed.

Janus(T) and Janus(A) Format

These options allow you to convert TRAC Janus(T) or Janus(A) terrain files directly into new JCM 2.2 terrain files. These files come in several different formats depending on the version number and the Terrain Builder may or may not be able to read the files. The original format is a 200 by 350 cell terrain file, a more recent version uses 350 by 350 cells. JCM 2.2 terrain files made from these formats are exact copies of the Janus(T) files. There is also a 698 by 698 format which is used by TRAC as a master file from which smaller files are extracted. Appendix A has a more detailed description of each of these formats and how to tell which kinds you may have.

This option is invoked from the Direct File Conversion menu by entering the **B** or **C** command and pressing **RETURN**.

TERBUILD asks:

Input Janus(T) Terrain File Number?

Enter a number in the range 1-999. Pressing {**PF4**} aborts the function. After the file number you need to enter the file type:

Janus(T) Terrain File Type?

Enter **1** for a 200 by 350 cell file, **2** for a 350 by 350 cell file, or **3** for a 698 by 698 cell file. Again, the {**PF4**} key may be used to abort this function.

From this point on TERBUILD behaves exactly the same as if you had read in a ARAC file.

User Format Files

This is the "hook" which allows the user to process files of any format not already provided by TERBUILD. The user must provide a routine called DIRECTLY_CONVERT_USER_FILE and relink the Terrain Builder program.

Convert Lat/Long to UTM

This function may be used to convert coordinates expressed in Latitude and Longitude to UTM coordinates. When invoked, the following menu is displayed:

```
        Convert Standard Lat/Long to UTM

                         Latitude        Longitude

     Lower Left Corner    0 : 0 : 0 N    0 : 0 : 0 E

UTM  coordinate:     nnnn.nnn,  nnnn.nnn,  Zone  nn


        Enter value to convert and hit RETURN.
             Hit PF4 to return to Main Menu.
```

Figure 10 Convert Lat/Long menu

Enter the desired Latitude in degrees : minutes : seconds North/South. Enter the desired Longitude in degrees : minutes : seconds East/West.

When the desired coordinates have been entered, press the **RETURN** key. The appropriate WGS84 UTM coordinates will then be displayed along with the UTM zone to which the point belongs.

To convert another set of Lat/Longs, use the **TAB** and **BS** {**F12**} (backspace) keys to move to the appropriate field(s), change their contents and press the **RETURN** key again.

Press the {**PF4**} key to **RETURN** to the Main menu.

Exit TERBUILD

The **X** command is used to terminate the Terrain Builder. If there is a terrain file in memory then you will be given one last opportunity to save it to disk before the program terminates.

Error Report File

When TERBUILD detects data errors it reports them in the file ERRFILE.LIS in the default directory. When the first error is detected the following message is displayed on the terminal:

Opening Error Record File.

TERBUILD detects and reports on two types of errors. The first type of error occurs when it finds that a desired elevation value in the source file is NULL. When this happens the error is reported in the error file:

NULL value encountered in scanline nnnnn, element eeee

The second type of error occurs when TERBUILD tries to move an entry from the source file into the terrain file and finds that the terrain file already has a value in that terrain cell. If the value differs from the value taken from the source file then an error is reported:

Point at (XXXX.XXX, YYYY.YYY):

MAPXY(iii,jjj) = nnnnn, Source = mmmmm

where:

(XXXX.XXX, YYYY.YYY) is the UTM coordinate of the point in question

MAPXY(iii,jjj) is the value currently in the terrain file in the located at (iii,jjj)

Source is the value from the source file

These errors are warning errors only. They do not cause TERBUILD to terminate.

Fatal Errors

Certain run-time errors will cause TERBUILD to abort its current function and **RETURN** to the previously displayed menu. These errors are reported to the user's terminal:

Unexpected end-of-file in data file

I/O error nnn opening file
I/O error nnn reading file
I/O error nnn closing file

Note: "nnn" is the VAX FORTRAN I/O error number (see the FORTRAN User's Guide).

User Modifications

It is intended that users be able to modify and extend TERBUILD themselves. In order to facilitate this, several additional files have been included.

TERBUILD as delivered only requires that your installation have a run-time license for FMS. If you want to extend TERBUILD with new features, modify or add FMS forms, you will also have to have the FMS application development license.

Command Files

You have been provided with two DCL command files. ASSIGN.COM is used to define the logical names and symbols used to compile, link and run the Terrain Builder. ASSIGN first invokes the JCM LOGIN.COM, which is part of the JCM shell, to define the JCM system logical names and symbols. It then defines a few more for use by TERBUILD itself and finally it defines names needed to recompile and relink. The logical name TAPEFILE within ASSIGN equates to the tapedrive name. Please check the name of the tapedrive before invoking ASSIGN.

The second command file, CLTERBUILD, is used to recompile source files and then relink the Terrain Builder. It is invoked as:

@CLTERBUILD list-of-files

where "list-of-files" is the optional list of FORTRAN source files which are to be recompiled before TERBUILD is relinked.

Other Files

You have also been provided with all of the FMS forms and the forms library used by TERBUILD, all the FORTRAN source files for the TERBUILD routines, and all the JCM system object libraries needed to link TERBUILD.

File extensions indicate what type of file they are:

Extension	**Type**
.COM	DCL command files
.DOC	This document
.EXE	The TERBUILD executable program
.FLB	FMS forms library (contains copies of the .FRM files)
.FOR	TERBUILD FORTRAN source files - compiled versions are stored in TERBUILD.OLB
.FRM	FMS forms
.FTN	FORTRAN source files for routines which are called by FMS - compiled versions are stored in FMS$LIB.OLB
.GBL	Global common blocks used by the .FOR files
.OLB	Object file libraries

Only the source files needed to maintain and modify TERBUILD have been included. The source files used to create the following object libraries have not been included:

CSCLIB.OLB - JCM system utility routines
IOCOM.OLB - Input/Output utility routines
TAPEIO.OLB - Tape drive control routines
VTSCREEN.OLB - VT-100 terminal control routines

Extending TERBUILD

There are two ways of extending TERBUILD to include new file types.

The first is to include them via the USER file "hooks". There are three routines which comprise the "hooks." They are:

CONVERT_USER_FILE - extracts data from one or more user files and loads it into a JCM file

DIRECTLY_CONVERT_USER_FILE - creates a JCM file which is an exact copy of a single user file

TRANSFER_USER_FILE - copies user's files from disk to tape

Dummy versions of these routines have been provided. You should replace them with your own routines and relink TERBUILD.

The FMS shell for TERBUILD has been constructed so that you can create the USER routines without having to access or use FMS. You may also create the USER routines with their own FMS shells as long as you remember to deactivate your FMS shell before returning to TERBUILD's shell.

The second way to extend TERBUILD is to add new features and file formats to any of TERBUILD's menus. This requires that you have a knowledge of FMS and the complete application development license for FMS. The program is designed to be easily modifiable, but if you need any help feel free to contact:

JCM Project Leader
Lawrence Livermore National Lab
P.O. Box 808; L-315
Livermore, CA 94550

510/423-4737

Appendix A

TERBUILD can read and process six different file formats. They are:

ARAC	- Atmospheric Release Advisory Committee
DTED	- Digital Terrain Elevation Data from the Defense Mapping Agency (tape and CD)
Janus (A)/(T)	- The formats used by TRAC for Janus(A) and Janus (T), their version of Janus
Janus 4.X	- An older version of LLNL Janus.
JCM1.x/Janus 5.1	- The previous version of JCM and the last version of LLNL Janus.
JCM 2.X	- JCM 2.2, the format used by all versions JCM 2.X

Each of these formats is more fully described in the following sections.

ARAC Format

ARAC files are made up of a series of card-image, ASCII records. The file is divided up into two sections, the file header section and the data file.

File Header Section

The file header section describes the area covered by the file. The important fields are the lower left and upper right corners and the file resolution fields. Following is a typical ARAC file header section:

AVERAGED FILE NAME: TFCHA
RECTANGLE UTMS (KM)

	E	N	
SW	382.00000	3889.00000	(lower left corner)
NW	382.00000	3915.00000	
NE	408.00000	3915.00000	(upper right corner)
SE	408.00000	3889.00000	

CENTRAL MERIDIAN = 93.0

DELX = 0.065000 DELY = 0.065000 (resolution = 65 meters)
IMAX = 400 JMAX = 400 (400 by 400 cells)

ACCIDENT COORDINATES:

UTMS		LAT	LONG
395.00000	3902.00000	35 15 26N	94 9 15W

The important descriptors are displayed on the user's terminal when the file is read:

ARAC file:

Upper Right Corner: xxxx.xxx, yyyy.yyy
Lower Left Corner: xxxx.xxx, yyyy.yyy

Resolution: nn meters, iii x jjj cells

Data File

The data file section is made up of blocks of data. Each block contains the elevation data for a single horizontal scanline. The file described above would have 400 blocks or scanlines, each containing 400 data points.

Each block begins with an 80 byte descriptor record which tells the block number and the UTM Y coordinate of the scanline. It is followed by a set of 80 byte records. Each record contains ten 8-byte data fields, one for each elevation point. The number of records in the set depends on the number of elements per scanline:

Number_of_Records = Number_per_Scanline / 10

If the number of elements per scanline is not divisible by ten, then the last record in the set will be short and the extra fields will be padded with spaces.

The X coordinate of each data point can be calculated by:

X = South_West_Easting + (Field_Number-1)*DELX

The Y coordinate of each element in a scanline can be calculated by:

Y = South_West_Northing + (Block_Number-1)*DELY

DTED Tape Format

The DMA DTED file consists of a series of data records which contain the digitized terrain elevation values. The file begins with a Volume Label and a Header Label followed by a User Header Label. The User Header Label describes the area contained within the file and the resolution of the data elements. If the source file is on tape then the User Header Label is followed by a tape mark.

The data records are organized as a series of vertical scanlines. Each element in the scanline indicates the elevation at the corresponding Latitude and Longitude. The number of the scanline determines the Longitude, the number of the element itself determines the Latitude. If the source file is on tape then there is a tape mark after the last data record.

Each file is terminated with an End-of-File Label, followed by a User Trailer Label. Again, if the source file is on tape, the labels are followed by a tape mark.

If the source file is on tape then there may be more than one file on the tape. If so, the next file begins with a new Header Label. There is an additional tape mark after the last file on the tape.

In summary tape file organization is:

```
Tape Volume Label        \
Header Label              \
User Header Label          \
<tape mark>       \
Data Records                  >  complete data file
<tape mark>       /
End-of-File Label          /
User Trailer Label        /
<tape mark>  /
Header Label             \
User Header Label         \
<tape mark>      \
Data Records                  >  complete data file
<tape mark>       /
End-of-File Label          /
User Trailer Label        /
<tape mark>  /

          • • •
```

```
Header Label          \
User Header Label      \
<tape mark>             \
Data Records             > complete data file
<tape mark>             /
End-of-File Label      /
User Trailer Label    /
<tape mark>          /
<tape mark>         /
```

DTED files which have been written to disk have the same format except that the tape marks have been removed:

```
Tape Volume Label      \
Header Label            \
User Header Label        \
Data Records              > complete data file
End-of-File Label        /
User Trailer Label      /
```

Note: The orginal Tape Volume Label is included in the disk file to allow trace back to the original source tape if necessary.

Tape Volume Label

The Tape Volume Label is 80 bytes long. The only part of the label that TERBUILD checks is the first four bytes. They must contain the characters "VOL1". The Volume Name from the label is displayed in the "Tape Label" field on the terminal as part of the description of the DTED file.

Header Label

The Header Label is 80 bytes long. When read, its contents are displayed on the terminal as part of the description of the DTED file. These fields are the File name, Reel number, File number, Version number, and the date the file was originally created.

Note: The creation date is the day of the year on which this file was created so it has a value in the range 1 through 366.

User Header Label

The User Header Label is 80 bytes long. It contains the information about the area contained within the DTED file. When read, its values are displayed on the user's terminal.

Note: The Classification Codes are:

T = Top Secret
S = Secret
C = Confidential
R = Restricted
U = Unclassified

File Descriptor Display

Whenever a DTED file is opened the following file descriptor display is printed on the user's terminal:

Tape File Labels:
Tape Label: tttttt Filename: nnnnnn
Reel #: nn File #: nn Version: nn Created: nnn, 19nn

User Label Information:
Classification Code: a
Corner at: dd:mm:ss a dd:mm:ss a
nnnn Longitude Lines nn.n seconds apart
nnnn Latitude Lines nn.n seconds apart

End-of-file Volume Label

The End-of-File Label is 80 bytes long. The only part of the label that TERBUILD checks is the first four bytes. They must contain the characters "EOF1".

User Trailer Label

The User Trailer Label is 80 bytes long. The only part of the label that TERBUILD checks is the first four bytes. They must contain the characters "UTL1".

Run-time Errors

TERBUILD detects and reports several types of run-time I/O errors. These errors are fatal and cause TERBUILD to abort the current function and **RETURN** to the previously displayed menu screen. These errors are:

No volume label
No file header label
No user header label
No end-of-file trailer label
No user trailer label

DTED CD Format

Refer to Military Specification Digital Terrain Elevation Data (DTED), MIL-D-89000, 1 October 1990.

JCM 2.2 Format

The JCM terrain file is stored in binary format. Eight data items are stored for each terrain cell. JCM currently allows up to 1000 x 1000 cells to be used. The data items are:

Elevation (lower left corner)
Tree/City/Water present (Boolean to indicate the presence of trees, urban, or water)
Tree/City density (uniformly distributed throughout cell) or Water depth (over entire cell)
Road (Boolean to indicate the presence of a road)
River (Boolean to indicate the presence of a river)

Tree/City blown down/rubbled (Boolean, this data is changed dynamically during a simulation, and is not changed by the Terrain Editor)
Tree/City on fire (Boolean, this data is changed dynamically during a simulation, and is not changed by the Terrain Editor)
Military Obstacle Present (Boolean, this data is changed dynamically during a simulation, and is not changed by the Terrain Editor)

Trafficability (not currently implemented)
Roughness (not currently implemented)

The Terrain Editor does not allow modification to elevation data, or data not implemented. It does allow input and modification to rivers, roads, trees (vegetation), cities (urban areas), and areas covered by water.

The binary file, organized as a sequential file, stores five sets of information (records).

The first set of information consists of the terrain file descriptors:

Lower left UTM	Floating	Two words: X and Y Coordinates
File width	Floating	Width of file in kilometers
File height	Floating	Height of file in kilometers
Grids wide	Integer	How many grid cells wide the file is (max 1000)
Grids tall	Integer	How many grid cells tall the file is (max 1000)
Upper right UTM	Floating	Two words: X and Y Coordinates
Width of cell	Floating	Cell width in kilometers
Height of cell	Floating	Cell height in kilometers
Inverse of width	Floating	One divided by cell width
Inverse of height	Floating	One divided by cell height
Latitude of Lower Left	Floating	Three words: degrees, minutes, seconds
Longitude of Lower Left	Floating	Three words: degrees, minutes, seconds
Meter conversion	Floating	Converts elevations to meters
Kilometer conversion	Floating	Converts elevations to kilometers
Feet conversion	Floating	Converts elevations to feet
Base elevation	Floating	Actual altitude of lowest point in file
Probability LOS blocked	Floating	Eight words: corresponding to density index (see **density index** in set four below)
Height	Floating	Eight words: corresponding to density index (see **density index** in set four below)
Trafficability	Floating	Eight words: corresponding to trafficability index (see **trafficability index** in set four below)
Roughness	Floating	Eight words: corresponding to micro-terrain roughness index (see **roughness index** in set four below)
Depth	Floating	Eight words: corresponding to Water depth (see **density index** in set four below)
Number River nodes	Integer	How many River nodes have been defined (max 2000)
Number Road nodes	Integer	How many Road nodes have been defined (max 4000)
File Area	Character	Eighty characters: file descriptive string

File Date	Integer	Three words: date file was first created (year, month, day)
File Time	Integer	Three words: time file was first created (hours, minutes, seconds)
File Source	Integer	Thirty-two bits: bits indicate which sources were used to generate the original file
File Meridian	Floating	Three words: Longitude used as the central meridian for the UTM or UTM-like coordinate system for the map (degrees, minutes, seconds)
Future expansion	- - - -	Twenty-five words: set aside for future use

The second set of information consists of the list of river nodes (number of nodes indicated by "Number River nodes" field above:

River node	Floating	Three words per node: X, Y, and flag (+1.0 = end of string, 0.0 = node connects to next node, -1.0 = last node on list)

The third set of information consists of the list of road nodes (number of nodes indicated by "Number Road nodes" field above:

Road node	Floating	Three words per node: X, Y, and flag (+1.0 = end of string, 0.0 = node connects to next node, -1.0 = last node on list)

The fourth set of information is the actual terrain data. To define an area of ***m*** by ***n*** data cells, it is necessary to use ***m+1*** by ***n+1*** scan, or grid lines. Each data point on a scan line contains several bit-packed fields, the ones of primary interest being the elevation at the data point, the tree/city density present in the grid cell (whose lower left corner is the data point), and the presence or absence of a river/road in that grid cell. The bit-packed fields which describe the terrain are organized into 32 bits per data point as follows:

0 - 15	elevation (16 least significant bits)
16	river present in cell flag
17	military obstacle present in cell flag
18	contents of cell on fire flag
19	contents of cell blown down flag
20	road present in cell flag
21 - 22	trafficability index (0-3; not currently used)
23 - 25	roughness index (0-7; not currently used)
26 - 28	density index (0-7; concealment and height for trees and

	urban areas, depth for water areas)
29	cell is urban area flag
30	cell is covered by trees flag
31	cell is covered by water flag

The fifth set of information contains the river search frames. The individual line segments which make up the rivers are allocated to different search frames based on their location relative to the edges of the terrain file. This technique is used to speed up the search when trying to determine whether or not a unit is trying to cross a river whenever it moves.

Schematically, the binary format terrain file looks like this:

File Descriptors
River Block
Road Block
Terrain Block
Search Frames

Appendix B

Reading Binary Tapes on the VAX

The VAX memory addressing scheme is not set up very well for binary tape processing. The VAX organizes its memory by byte addresses, each byte can be addressed separately. Entities which consist of more than one byte are addressed by their low order byte. So if a 32 bit word has the hexadecimal value '12345678' it is stored in memory as '78563412'. This is not a problem if all fields are multiples of 8 bits long and byte displaced. However, if the entities are not so organized, it becomes very hard to access them, even with the bit twidding routines provided in the VAX Run-Time Library.

The digitized terrain files are organized as a continuous stream of bits. In order to read and access such data two things must be accomplished:

1. Read the file into a BYTE structured array, but read it backwards. For example, if the data is 100 bits long then define a BYTE array which is 13 bytes long (100 bits divided by 8 bits per byte equals 12.5 bytes which rounds up to 13 bytes). In order to read the binary information:

```
BYTE ARRAY(13)
...

CALL LIB$GET_LUN (LUN)


OPEN (UNIT=LUN, NAME='file or tape name', TYPE='OLD',
BLOCKSIZE=13)


READ (LUN,1001) (ARRAY(J), J = 13, 1, -1)
1001 FORMAT (13A1)
```

This loads the binary bits into memory in the proper order.

2. Extract the bits from the high order end back. From the above example, if the entities are 10 bits long then the entity numbered INDEX is accessed by:

```
INTEGER START, ENTITY, INDEX
...

START  =  100 - (10*INDEX)
```

```
ENTITY = LIB$EXTZV(START,10,ARRAY)
```

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551